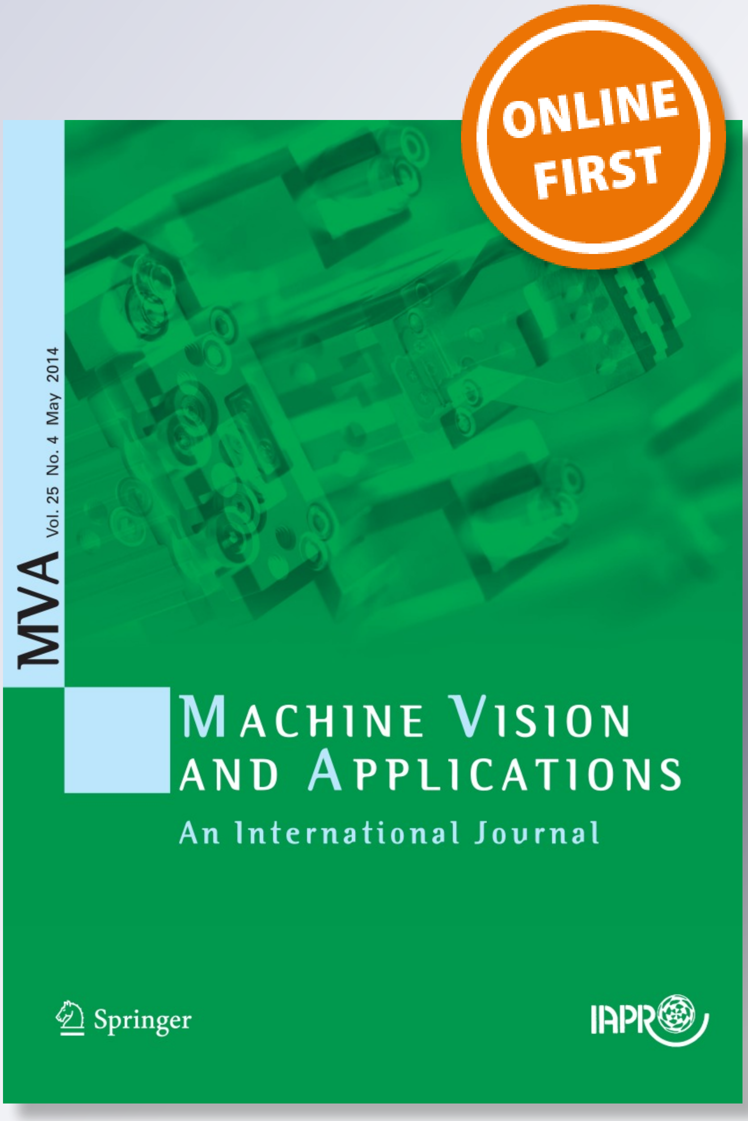
Graph-cut based interactive segmentation of 3D materials-science images

Jarrell Waggoner, Youjie Zhou, Jeff Simmons, Marc De Graef & Song Wang

Machine Vision and Applications

ISSN 0932-8092

Machine Vision and Applications
DOI 10.1007/s00138-014-0616-3



Report Documentation Page		*Form Approved* *OMB No. 0704-0188*

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **2014**	2. REPORT TYPE	3. DATES COVERED **00-00-2014 to 00-00-2014**
4. TITLE AND SUBTITLE **Graph-cut based interactive segmentation of 3D materials-science images**		5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Materials and Manufacturing Directorate,,Air Force Research Labs,Dayton,OH, 45433**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR’S ACRONYM(S) 11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**18**	

Machine Vision and Applications
DOI 10.1007/s00138-014-0616-3

ORIGINAL PAPER

Graph-cut based interactive segmentation of 3D materials-science images

Jarrell Waggoner · Youjie Zhou · Jeff Simmons · Marc De Graef · Song Wang

Received: 10 November 2013 / Revised: 10 March 2014 / Accepted: 27 March 2014

Abstract Segmenting materials' images is a laborious and time-consuming process, and automatic image segmentation algorithms usually contain imperfections and errors. Interactive segmentation is a growing topic in the areas of image processing and computer vision, which seeks to find a balance between fully automatic methods and fully-manual segmentation processes. By allowing minimal and simplistic interaction from the user in an otherwise automatic algorithm, interactive segmentation is able to simultaneously reduce the time taken to segment an image while achieving better segmentation results. Given the specialized structure of materials' images and level of segmentation quality required, we show an interactive segmentation framework for materials' images that has three key contributions: (1) a multi-labeling approach that can handle a large number of structures while still quickly and conveniently allowing manual addition and removal of segments in real-time, (2) multiple extensions to the interactive tools which increase the simplicity of the interaction, and (3) a web interface for using the interactive tools in a client/server architecture. We show a full formulation of each of these contributions and example results from their application.

Keywords Image segmentation · Materials volume segmentation · Segmentation propagation · Interactive segmentation · Graph-cut approaches

1 Introduction

Interactive segmentation is a rapidly growing area of computer vision and has seen heightened interest recently [28,49]. While traditional segmentation seeks to identify objects/structures within an image in a fully automated fashion, interactive segmentation, similar to active learning [43], accomplishes the goal of image segmentation while incorporating a sparse number of user interactions which are included as additional constraints or guidance in the segmentation model or algorithm. These interactions may take on different forms, and may include drawing a bounding box [39], roughly outlining a boundary [32], or drawing brush strokes inside and/or outside the object of interest [3,41,54,57]. A desired property of an interactive segmentation approach is that the user interaction be as convenient (i.e., low cognitive load) and sparse (i.e., few in number) as possible, while simultaneously providing immediate feedback to the user on every interaction.

One domain that has been unaddressed in interactive segmentation literature is materials science image segmentation, where there are no existing techniques focusing solely on segmenting materials' images using an interactive approach. Materials science is especially important to the development of new metals and biomaterials, and presents unique challenges in image segmentation. First, materials' images often are 3D volumes [21] made up of a sequence of individual 2D image "slices," as shown by the two sample slices

Electronic supplementary material The online version of this article (doi:10.1007/s00138-014-0616-3) contains supplementary material, which is available to authorized users.

J. Waggoner · Y. Zhou · S. Wang (✉)
University of South Carolina, Columbia, USA
e-mail: songwang@cec.sc.edu

J. Waggoner
e-mail: waggonej@email.sc.edu

J. Simmons
Materials and Manufacturing Directorate,
Air Force Research Labs, Dayton, USA

M. De Graef
Department of Materials Science and Engineering,
Carnegie Mellon University, Pittsburgh, USA

Published online: 26 April 2014

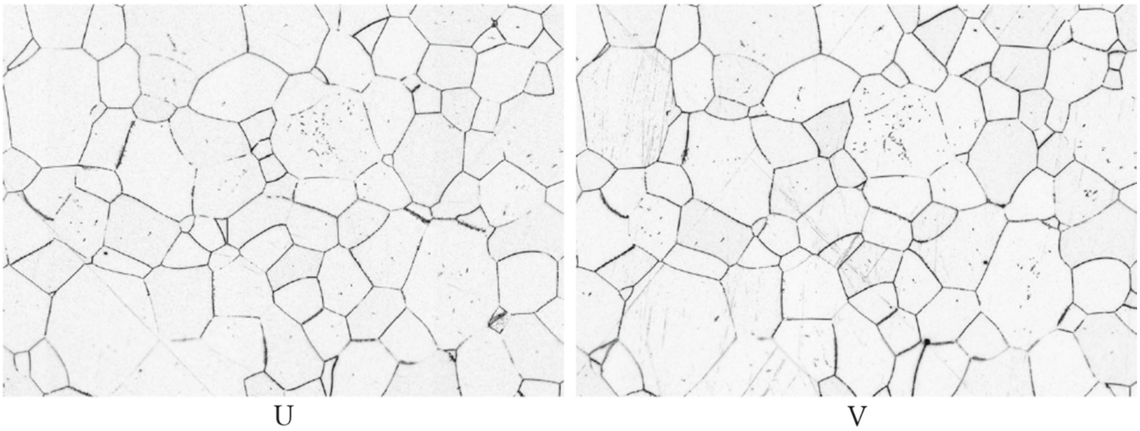


Fig. 1 Two adjacent slices of a titanium image volume [40]. Image intensity inverted for clarity; the original is similar in appearance to Fig. 11

in Fig. 1 [1]. This large number of slices must all be segmented to fully and properly analyze the 3D structure of the material. Second, depending on the inter-slice distance, the 2D structure in two neighboring slices may show high continuity. Such inter-slice structure continuity must be considered to achieve accurate segmentation. Third, materials volumes consist of numerous substructures (e.g.,"grains" in a metallic material, or "cells" in a biomaterial, etc.) with complex relationships (e.g., adjacency/nonadjacency relationships) among them that determine many desirable properties of the material [38,51]. Existing interactive segmentation techniques often only focus on foreground-background segmentation [3,39], and may not scale to the large number of substructures present in materials' images. Other methods may handle multiple structures [49,50], but do not incorporate any prior knowledge about the unique relationships among substructures in materials' images [37,52]. Finally, the imaging techniques used to obtain a materials image volume may result in significant noise or other ambiguities that increases the difficulty to segment a materials image volume in a fully automatic fashion. Many of these properties are not unique to materials' images, e.g. medical images may exhibit complex relationships, neuroimages may have high continuity between slices, natural images may contain numerous structures, etc. and as such, advances in interactive segmentation may have far-reaching consequences.

In this paper, we present an interactive segmentation approach to segment materials science image volumes. We show that an existing propagation-based materials image segmentation approach [58] can be extended to allow for convenient interactive segmentation. We illustrate the performance of the proposed approach by using it to segment a materials image volume using smaller number of interactions compared with general-purpose interactive segmentation methods that do not incorporate materials-specific priors. Further, we develop extensions to the interactive tools we present–namely, annotation repropagation, parameter estimation, and salient region detection extensions–and show that these extensions lead to improved performance. Finally, we illustrate the client/server architecture used to build a web application for implementing the proposed approach and related extensions.

1.1 Related work

Though no existing work focuses on the materials image segmentation application using interactive segmentation, we discuss related work that focuses on either of these two aspects.

There are a number of existing, non-interactive approaches to segment materials' images [10,48]. Among the most prominent is the work of Comer et al. [12,13] on the EM/MPM algorithm, originating from [30]. Other methods that have been specifically used on materials' images include graph cut [20,58], stabilized inverse diffusion equations [19], Bayesian methods [11,47], and the watershed [29] method. Most often, materials' images are opportunistically segmented by the simplest tools available, such as thresholding [17,45], or out-of-the-box methods such as watershed or normalized cut [46]. However, these methods do not incorporate any interaction for manual refinement by a user. Some of these approaches may require significant time to run; requiring the user to examine and correct errors only after the algorithm is complete may not be practical if rapid refinement is desired. Conversely, the general-purpose interactive segmentation techniques discussed previously do not incorporate any specific domain knowledge about materials' images, and thus may require additional effort on the part of the user than may otherwise be needed when segmenting a materials' image volume.

[1] For clarity, we inverted the image intensity in this figure, as well as several other figures in the later sections. The original is similar in appearance to Fig. 11.

General interactive methods exist to segment the object of interest using a model learned from user interactions [3,39, 54]. Other approaches incorporate interaction into morphological operations (watershed) [49], co-segmentation [1], or incorporate machine-learning to aid in the interactive process [28,53]. These interactive methods have been applied to a number of domains, including natural images [39], medical images [4], and neuroimages [49,50].

Interactive methods that focus on segmenting 3D volumes share elements in common with the proposed work. The majority of these related works are in the medical imaging area. [18] minimizes an energy function, though the formulation is different from the proposed approach. Both [9,53] employ a GPU implementation, however [9] includes a learning component, as we do in Sect. 7. Most similar to the proposed method, [33] uses a propagation framework to do 3D interactive segmentation on medical images. A graph cut-based minimization is used by [2], as is used in this paper. Finally, [22] introduce a web interface for interactive segmentation, as we do in this paper. All of [2,9,18,22,33,53] are specific to medical applications, and thus are targeted at a single or small number of segments, and are not able to handle large numbers of segments with different adjacency relations, as we do in this paper.

There are also a number of general 3D interactive methods that are not specific to medical images. Among these is [26], which is designed specifically to divide an existing 3D mesh, and does not apply to the serial-sectioned 3D volumes used in this paper. Further, [25] builds a tool-focused framework that includes “hole filling,” “point-bridging,” and “surface-dragging” tools. We similarly develop a tool-based approach along with multiple extensions, but we use a different approach (addition and removal) for the interaction in this paper. Finally, most similar to the materials application in this paper, the proprietary INTERSEG [35] plugin for ImageJ [42] focuses on interactively segmenting cell-like structures. These structures, however, have different types of adjacency relations compared with the grain-structures discussed in this paper, and thus the INTERSEG plugin uses different tools for interaction that we introduce in Sects. 3 and 4.

The remainder of this paper is organized as follows: In Sect. 2 we discuss the background for the proposed interactive segmentation approach for materials image volumes. We further discuss how the proposed approach handles segment removal in Sect. 3 and segment addition in Sect. 4. The first extension of the proposed method in Sect. 5 discusses how adjacent slice similarity can be leveraged to allow for segment repropagation to reduce the number of interactions required. In Sect. 6, we show how some of the parameters of the proposed method can be automatically estimated. The final extension of the proposed approach in Sect. 7 shows how a simple online learning system can detect salient regions that should be given extra attention by a human annotator. In Sect. 9, we evaluate the proposed method's performance against another general-purpose interactive segmentation method. Finally, in Sect. 10 we provide brief concluding remarks.

2 Interactive materials image segmentation

In [58] we developed a 3D materials science image segmentation method by propagating segmentation S^U of a slice U to a neighboring slice V, resulting in a segmentation S^V. This way, using an initial segmentation on one slice, we can repeatedly propagate this segmentation to the remaining slices in the volume to obtain a complete 3D segmentation. This propagation was done while preserving the topology (i.e., non-adjacency relations among 2D segments), which led to a better performance when compared with methods that did not incorporate topology as a prior. Specifically, let the segmentation

$$S^U = \{S_1^U, S_2^U, \ldots, S_n^U\},$$

where $S_i^U, i = 1 \ldots n$ are disjoint segments in slice U, and this collection of segments makes up a partition of the slice U,

$$U = \bigcup_{i=1}^{n} S_i^U.$$

An example is shown in Fig. 2 where all the segments (“grain” structures) are separated by red lines. To propagate segmentation S^U to a new slice V to yield the segmentation S^V, we minimize the energy

$$E(S^V) = \sum_{p \in V} \Theta_p(S_i^V) + \sum_{\{p,q\} \in \mathcal{P}_n^V} \Phi_{pq}(S_i^V, S_j^V) \tag{1}$$

where $\mathcal{P}_n^V$ is the set of all four-connected pixels in V. The unary term $\Theta_p(S_i^V)$, which represents a cost for a pixel p being assigned to a segment S_i^V in slice V, was set to reflect the structure continuity between U and V,

$$\Theta_p(S_i^V) = \begin{cases} 0, & \text{distance}(p, S_i^U) < d \\ \infty, & \text{otherwise} \end{cases} \tag{2}$$

where d is a dilation distance that reflects the maximum possible structural change between U and V [58]. In addition, the binary term $\Phi_{pq}(S_i^V, S_j^V)$, which represents a cost for a pair of neighboring pixels p, q being assigned to two (possibly the same) segments S_i^V, S_j^V, was constrained to preserve non-adjacency segment relationships from U to V; i.e., any two segments S_i^V, S_j^V are allowed to be adjacent (have pixels that are four-connected between them) only if the corresponding segments S_i^U, S_j^U are also adjacent,

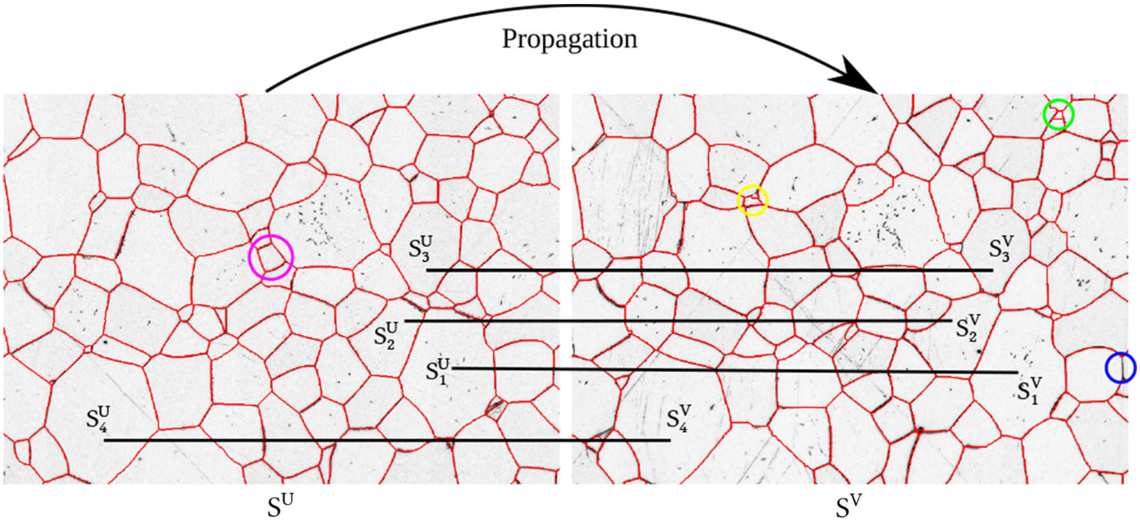


Fig. 2 Example of segmentation propagation, highlighting different types of topology changes, including segments that become unadjacent in V (S_2, S_3), segments that disappear in V (*purple*), segments that appear in V (*yellow*), an erroneously missing segment (*blue*) and a spurious segment (*green*). Image intensity inverted for clarity

$$\Phi_{pq}(S^V_i, S^V_j) = \begin{cases} 0, & i = j \\ \infty, & \{S^U_i, S^U_j\} \notin \mathcal{A}^U \\ g(p,q), & \{S^U_i, S^U_j\} \in \mathcal{A}^U \end{cases}, \tag{3}$$

where $\mathcal{A}^U$ contains segment pairs that are adjacent in S^U, and we set $g(p,q)$ to reflect the image boundary information in V [58]. An example is shown in Fig. 2, where S^V_1 and S^V_2 are allowed to be adjacent because S^U_1 and S^U_2 are adjacent in S^U. However, S^V_1 and S^V_4 are not allowed to be adjacent (have an infinity penalty) because S^U_1 and S^U_4 are not adjacent in S^U. This topology constraint was found to be particularly important for materials' images, and our proposed method was able to outperform other methods that did not incorporate such a prior.

While finding the global minimum to this cost is NP-hard, this cost has been shown to be minimizable to a local optimum using a graph-cut approach [6,55]. However, one phenomenon that was observed in this previous work was that, during propagation, 2D structure topology between U and V might not always be fully consistent. For example, a new 3D structure with no intersection in slice U might appear in slice V, e.g., the structure in the yellow circle in Fig. 2. Similarly, a 3D structure intersected by slice U might disappear in slice V, such as the structure circled in magenta in Fig. 2. This breaks the topology constraints given in Eq. (3) in some local regions. This may lead to spurious segments and missing structures, as circled in green and blue respectively, in Fig. 2.

Our method made use of a brute-force automated search to locate such spurious and missing structures in V [58]. Given a large number of substructures in a material sample, it is computationally expensive to examine every location for possible spurious or missing structures. Furthermore, given the small inter-slice distance, spurious or missing structures are usually very small and thus difficult for an automatic algorithm to segment correctly. In this paper, our goal is to develop effective interactive tools to allow a user to conveniently specify the local areas that contain spurious or missing structures, and incorporate such interactions to refine the segmentation S^V to a corrected $\tilde{S}^V$ on slice V, using the same energy minimization algorithm. More specifically, we propose to allow the user to correct these two types of segmentation errors within this segmentation propagation framework by: (a) annotating the location of a new segment to handle cases where a new structure appears in slice V, and (b) annotation of existing segments that should no longer be present in segmentation S^V.

These interactions are inherently local because the 2D cross section of a 3D structure shows very small size before appearing or disappearing from a neighboring 2D slice. Therefore, correcting S^V to $\tilde{S}^V$ can be achieved by using the same energy minimization in local image areas around the interactive annotations. This is also important because interactive segmentation requires instantaneous user feedback. The previous propagation method segmented entire slices, which was more computationally intensive than is desirable in an interactive system. We will further discuss these two interactions, and how we identify local regions for each, in the following subsections.

3 Removal of spurious segments

For this interaction, we allow the user to select a spurious segment S^V_k for removal by clicking the mouse on this segment in a visualized segmentation of S^V. Instead of naively

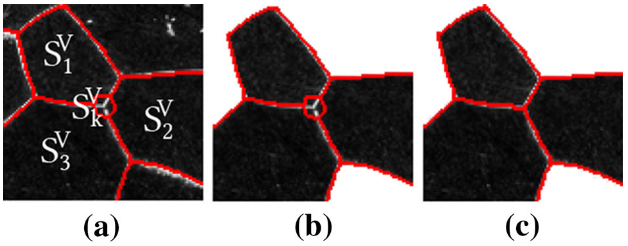


Fig. 3 Example selection of a spurious segment S_k^V for removal. **a** Chosen S_k^V and surrounding segments. **b** Local region extracted around S_k^V. **c** The updated segmentation in the extracted local region

removing this segment by arbitrarily merging it into one of its neighbors, we use the same energy minimization discussed above to assign the individual pixels in S_k^V to potentially different neighboring segments. As discussed above, we identify a local region in which we update the segmentation. Specifically, this local region consists of the specified S_k^V and its neighboring segments, e.g., S_1^V, S_2^V, S_3^V surrounding the spurious segment S_k^V in Fig. 3a, and re-run the energy minimization within this local region after modifying the Θ term to incorporate the interaction, resulting in an updated segmentation in this local region, as shown by the example in Fig. 3c. For ease of notation, we use similar notation to the adjacency definition in Eq. (3) by using $\{\mathcal{A}^V\}_k$ to refer to the set of segments neighboring the segment S_k^V. This way, the local region for updating the segmentation is

$$\mathcal{L} = \{\mathcal{A}^V\}_k \bigcup S_k^V. \tag{4}$$

In this local region, we rerun the energy minimization of Eq. (1) by modifying the Θ term. In particular, we do not allow any pixel to be assigned to S_k^V since this segment is to be removed. Instead, the pixels initially in S_k^V can be assigned to any of the segments in $\{\mathcal{A}^V\}_k$ with 0 cost for the Θ term, i.e.,

$$\begin{aligned} &\forall p \in S_k^V, \quad \Theta_p(\tilde{S}_i^V) = \begin{cases} 0, & S_i^V \in \{\mathcal{A}^V\}_k \\ \infty, & otherwise \end{cases} \\ &\forall p \notin S_k^V, \quad \Theta_p(\tilde{S}_i^V) = \Theta_p(S_i^V) \end{aligned} \tag{5}$$

By updating Θ in this fashion, we do not require the pixels previously in S_k^V merged into a single neighboring segment. Instead, these pixels may be assigned to more than one segment in $\{\mathcal{A}^V\}_k$, as shown in Fig. 3c.

Note that this interaction is very simple and convenient, as it requires only a single click anywhere inside the spurious segment S_k^V. The full algorithm for removing spurious segments is summarized in Algorithm 1.

4 Addition of missing segments

Unlike removal, interactively annotating an additional structure cannot be solely formulated as a simple modification of the Θ term in the energy minimization formulation. This is because the multi-labeling problem used to optimize the energy minimization form in Eq. (1) optimizes over a fixed set of segments, and cannot introduce new segments. Thus, for each missing segment, we must explicitly create a new segment at the location interactively specified by the user.

Based on the initial segmentation $S^V = \{S_1^V, S_2^V, \ldots, S_n^V\}$, we take as input from the user an annotation specifying the center location c of the new segment $\tilde{S}_{n+1}^V$. In addition to this, we also accept two parameters from the user: (1) the seed radius s specifying a circular region around c such that this circular region is completely contained within the missing structure; (2) a dilation radius d, which is similar to the dilation parameter used in Sect. 2, such that the circular region with this dilation radius d centered at c completely covers the missing structure to be segmented. We explicitly enforce that $d \geq s$ for any choice of s. We call pixels within the seed radius s of c "seed pixels" and pixels within the dilation radius d of c "dilation pixels." In this interaction, seed pixels are guaranteed to be part of the missing segment that is added, as shown by the green circle in Fig. 4b, and dilation pixels, excluding seed pixels, are potentially part of the missing segment that is added, as shown by the blue area in Fig. 4b. This makes the selection of s and d conceptually simple for the user to tune. In Sect. 6, we discuss how to automate the selection of s and d to further reduce the user's burden when interactively segmenting a materials volume.

Algorithm 1 Interactively specifying segment to remove.

```
function RemoveSegment($S^V$, $S_k^V$)
    $\mathcal{L} \leftarrow \{\mathcal{A}^V\}_k \bigcup S_k^V$
    For the pixels $\mathcal{L}$, build graph for energy minimization problem from Section 2
    $\Theta \leftarrow$ set from Eq. (5)
    $\tilde{S}^V \leftarrow S^V$ incorporating the updated segmentation in $\mathcal{L}$
    return updated $\tilde{S}^V$
end function
```

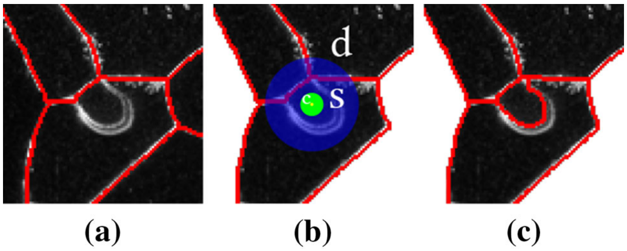


Fig. 4 Annotating the addition of a missing segment. **a** Segmentation S^V with a missing segment near the center of the image. **b** Annotation of a center point c, along with a seed radius s and a dilation radius d, and the identified local region for updating the segmentation. **c** The updated segmentation of the local region shown in (**b**)

Algorithm 2 Interactively specifying segment to add.

```
function AddSegment($S^V, c, s, d$)
    $\mathcal{L} \leftarrow$ union of all segments that contain a seed pixel or dilation pixel
    For the pixels in $\mathcal{L}$, build graph for energy minimization problem from Section 2
    $\Theta \leftarrow$ set from Eq. (6) and Eq. (7)
    $\tilde{S}^V \leftarrow S^V$ incorporating the updated segmentation in $\mathcal{L}$
    return updated $\tilde{S}^V$
end function
```

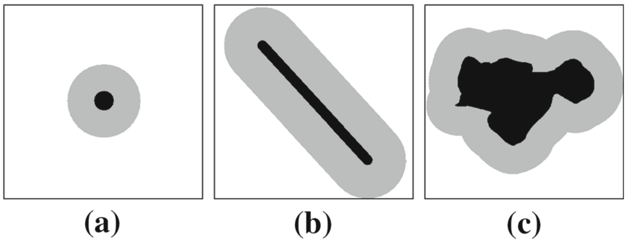


Fig. 5 Alternate annotation shapes for addition. **a** Standard point-annotation. **b** Line annotation. **c** "Scribble" annotation. *Dark* areas are the seed pixels and *grey* areas are the dilation pixels

Similar to the removal interaction in Sect. 3, we define a local region around the specified c to update the segmentation of S^V. Specifically, we define this region by taking all segments in S^V that contain one or more seed or dilation pixels. In this local region we modify the Θ term of the energy minimization in Eq. (1) to obtain an updated segmentation. Specifically, we allow all seed and dilation pixels to be reassigned to the new segment $\tilde{S}^V_{n+1}$ by setting

$$\Theta_p(\tilde{S}^V_{n+1}) = \begin{cases} 0, & \|p - c\| \leq d \\ \infty, & \text{otherwise} \end{cases} \tag{6}$$

where $\|\cdot\|$ is the Euclidean distance between pixels p and c. Furthermore, to ensure that the seed pixels are always guaranteed to be part of $\tilde{S}^V_{n+1}$ we set an infinity penalty for seed pixels assigned to any segment other than $\tilde{S}^V_{n+1}$,

$$\Theta_p(\tilde{S}^V_i) = \begin{cases} \infty, & \|p - c\| \leq s \text{ and } i \neq n+1 \\ \Theta_p(S^V_i), & \text{otherwise.} \end{cases} \tag{7}$$

The full algorithm for adding a missing segment is summarized in Algorithm 2.

Note that annotations need not be constrained to a single point, as shown in Fig. 5a. Line-based annotations, as shown in Fig. 5b can be defined by setting seed pixels to be all those pixels within a distance s of any position along an annotated line. Dilation pixels can be defined similarly. Further, any "scribble"-like annotation, provided it remains connected and without holes, can be morphologically dilated by s or d to obtain seed or dilation pixels, respectively, as shown in Fig. 5c.

5 Annotation repropagation

While annotations defining addition or removal of segments can be made on a single slice, they may provide valuable information for adjacent slices and, as such, can be propagated (which we refer to as "repropagation") to these adjacent slices, similar to the segmentation propagation discussed in Sect. 2. For removal, we simply locate the same segment in all adjacent slices (if present), and repeat the removal operation summarized in Algorithm 1. For addition, illustrated in Fig. 6, we create the new segment as summarized in Algorithm 2, after which we repropagate the segmentation, including the newly created segment, to the adjacent slices in the same manner as discussed in Sect. 2. This repropagation can be done within the local region $\mathcal{L}$ on the adjacent slices for efficiency, and terminates if the new segment is no longer present during the repropagation (determined by the energy minimization discussed in Sect. 2).

When errors are small, such annotation repropagation may not be particularly beneficial. However, when there are gross errors, annotation repropagation can reduce the number of needed annotations significantly.

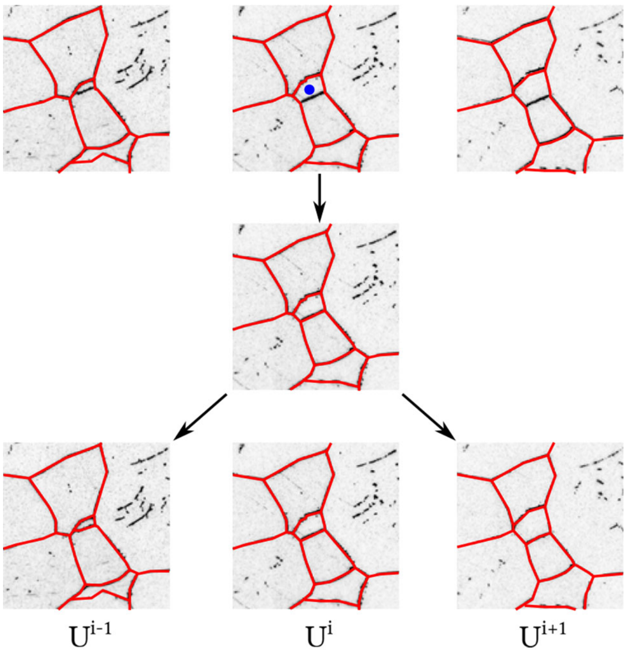


Fig. 6 Annotation repropagation for addition. The *blue* annotation on slice U^i (row 1) yields an updated segmentation of slice U^i (row 2). This new segmentation can be repropagated to the neighboring slices U^{i-1} and U^{i+1} (row 3) using the same propagation approach introduced in Sect. 2. Image intensity inverted for clarity

6 Parameter estimation

When interactively adding a new segment, as discussed in Sect. 4, the seed radius s and dilation radius d are required to be specified by the user. This results in additional burden on the part of the user. In this section, we develop a parameter estimation approach to automatically select these two parameters so the user need only override them in very rare cases, or not at all.

We do this by leveraging information about the center c the user provided relative to the initial segment in which it resides. Generally a missing segment occurs when 2D cross-section intersects a new 3D structure in V. Given a small inter-slice distance, as discussed in Sect. 2 where neighboring slices have large similarity, we expect that these missing segments are often small compared with its neighboring segments in slice V. An example is shown in Fig. 7a, where a small segment is missing (indicated by the yellow circle) in the segmentation S^V: this missing segment is mistakenly merged into a large neighboring segment S_b^V. Intuitively, placing c near the boundary of S_b^V likely indicates the missing segment is small, as shown by Fig. 7b. Conversely, placing c closer to the center of S_b^V likely indicates the resulting missing segment is large as shown in Fig. 7c. We make a simplifying assumption that we do not allow the missing segment to spill over the boundary of S_b^V. For example, the selection of c and s in Fig. 7b is able to generate the updated segmentation shown in Fig. 7d.

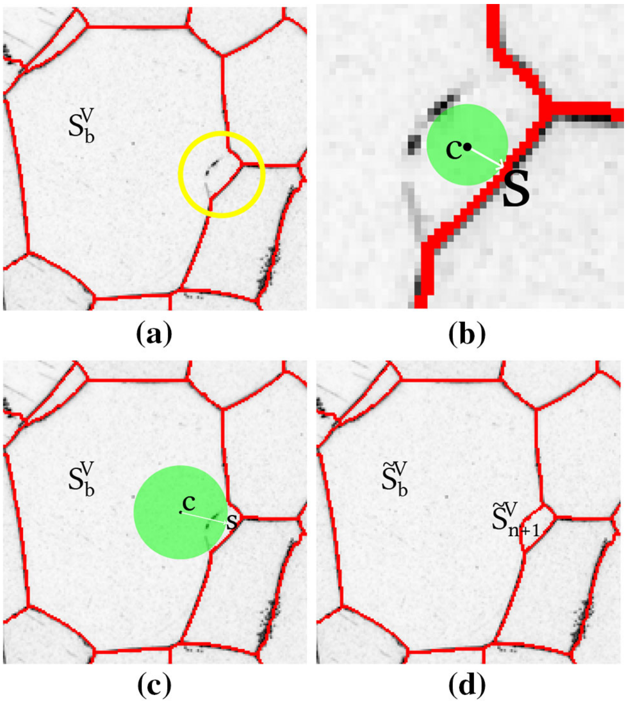


Fig. 7 Automatic selection of seed radius s and dilation radius d. **a** A missing segment located within a large segment S_b^V. **b, c** Selections of c at varying distances from the boundary of S_b^V, resulting in different estimations of s. **d** Updated $\tilde{S}^V$ by adding a missing segment using the seed center c shown (**b**) and the proposed parameter estimation method to determine s. Image intensity inverted for clarity

To obtain an estimation of s we start by setting $s = 0$, and we then increase s by a small ϵ amount until the circle centered at c with radius s is within ϵ distance of the boundary of the containing segment S_b^V, as shown by the arrow in Fig. 7b, c. In materials' images, the majority of newly appearing structures when moving from one slice to another are usually near the boundary of an existing structure S_b^V (near a "Y"-junction boundary between structures). This automatic approach is ideally suited for these cases. When the user specifies a c that falls directly on a segment boundary in S^V, we default to requiring user-supplied s in these less-common cases. For estimating d, it is scaled according to the value of s. Specifically, we set $d = 2 \cdot s$. As shown in Sect. 9, this approach saves both time and effort.

7 Salient region detection

Because materials' images can be very large and complex, it can take a significant amount of time for a human annotator to review the segmentation of such a large image to determine where it may require additional interaction. In this section, we introduce a salient region detection approach that identifies candidate regions highlighting the areas most likely to need additional interaction. In this work, we define a salient region as a subset of the complete image that is most likely to require interactivity, i.e. where the segmentation shows large uncertainty. Since the addition of missing segments is the most time-consuming annotation task, we focus on detecting the edges in the image that are not identified as segment boundaries, indicating a missing segment, for this salient region detection. As such, we identify prominent edges fragments that are not segmented during the propagation as candidate regions, and use a SVM classifier [14] to learn which candidates are truly salient regions, and which are noise that can be ignored by the human annotator. These salient regions are later enclosed in a bounding box for easier visualization.

More specifically, we use the online learning [44] system outlined in Fig. 8. We first use the Canny edge detector [8] in areas that are a fixed distance from the segmentation boundaries, preventing edges in the image which already fall on segmentation boundaries from being considered candidates, effectively leaving "residual" edges, which may correspond to either edges of missed structures or noise. The output of the edge detector is dilated slightly and then separated into connected components to produce candidate salient regions, which are then classified using an initial pool of 300 positive and negative samples to determine which should be presented as salient regions. Finally, when the user is presented

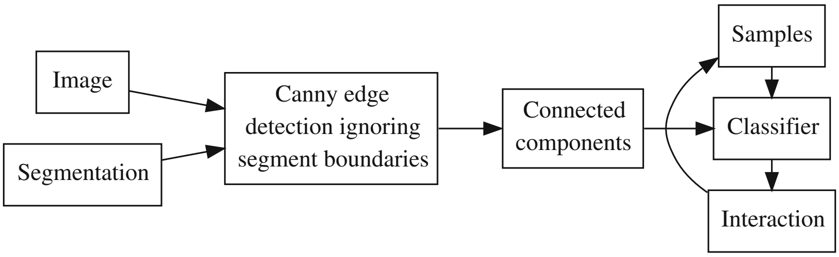


Fig. 8 Salient region detection pipeline. *Edges* are detected and classified as salient or non-salient. Those classified as salient are then shown in a *b*ounding box displayed on the image to guide the interactive segmentation. Based on the interaction, positive and negative samples can be extracted to enhance the classifier when used on other slices in the volume

with the salient regions highlighted on the segmentation, any subsequent interactive annotations are converted to positive samples, and remaining unannotated salient regions are converted to negative samples. This process is then repeated for the next image/segmentation pair, with updated samples integrated into the classifier.

To classify each candidate salient region, we extract a feature vector consisting of multiple shape and intensity properties, including

- The total area of the region,
- The minor and major axis length of the ellipse fit to the region,
- The maximum intensity inside the region, and
- The mean intensity inside the region.

These properties can be computed quickly, which fits well with the real-time requirements of the interactive segmentation problem. For the classifier, we use a SVM with a RBF kernel ($\gamma = 0.01$, $C = 1.0$), which can be retrained while a new image is being annotated. Sample salient region detection results are shown in Fig. 9, with true positives highlighted by blue bounding boxes and false positives highlighted by yellow bounding boxes. We can see that false positive detections of salient regions may occur when there are strong edge-like noises present in the image. The performance of this classifier on the initial segmentation of the Ti-21S dataset, trained on 128 previously recorded human annotations, is shown in Table 1, where FP, FN, TP, and TN represent false positives, false negatives, true positives and true negatives, respectively.

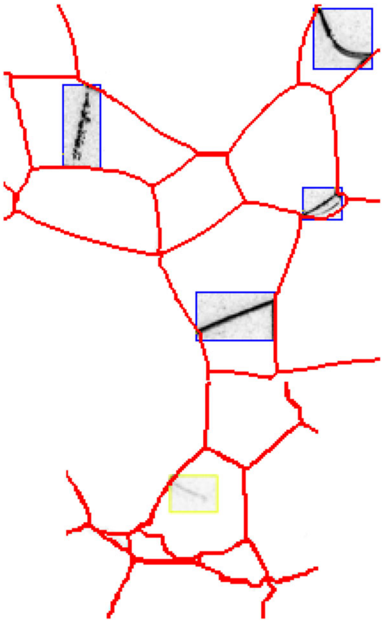


Fig. 9 Qualitative results of salient region detection. True positive detections of salient regions are surrounded by *b*lue bounding boxes. *R*ed curves are the detected segment boundaries before interactive segmentation. A false positive detection of salient regions is also shown surrounded by a *y*ellow bounding box. Image intensity inverted for clarity

8 Implementation

The energy minimization components, along with the approach from Sect. 2, are implemented in Python [36] using the NumPy/SciPy [24], scikit-learn [34], and OpenCV [7] libraries, along with the publicly available graph cut optimization (GCO) [56] library based on [5,6,27]. The interactive interface is built as a web application using the Django [15] web framework for the backend, and a custom single-page JavaScript client as the frontend. In the following, we discuss the internal architecture that enables interaction with the large images used in the proposed system, along with the developed interface.

8.1 Architecture

The proposed approach is implemented using a client/server architecture that allows different frontend interfaces to meet different needs for interaction. As shown in Fig. 10, our implementation consists of a large disk drive as a datastore, a

Table 1 Salient region detection classifier results on the initial segmentation of the Ti-21S dataset after training on 128 previously recorded human annotations

Slice	FP	FN	TP	TN	Precision	Accuracy
1	1	1	3	17	0.75	0.91
2	1	3	7	9	0.88	0.8
3	2	2	6	22	0.75	0.88
4	1	1	11	10	0.92	0.91
5	2	6	2	30	0.5	0.8
6	1	4	1	19	0.5	0.8
7	1	3	4	5	0.8	0.69
8	1	3	7	8	0.88	0.79
9	1	4	11	15	0.92	0.84
10	1	5	10	21	0.91	0.84
Total	12	32	62	156	0.84	0.83

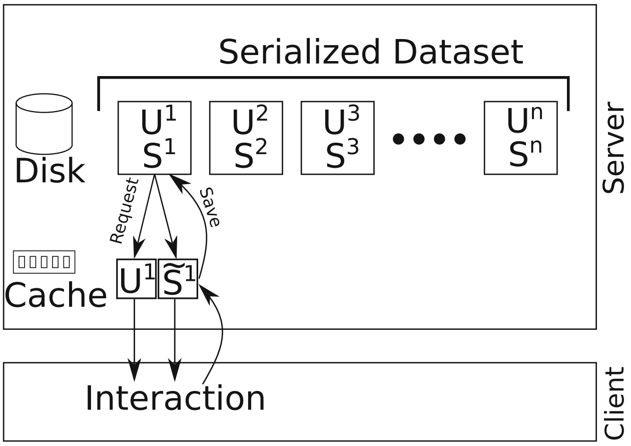


Fig. 10 Overview of the client/server architecture used to implement the proposed approach. Large datasets are persisted on disk with both the underlying image (U^i) and segmentation from the automatic propagation approach (S^i) saved for retrieval. A cache allows multiple interactions that modify the segmentation S^i to be saved in memory, where the image and segmentation can be quickly retrieved and modified. The client may explicitly issue a "Save" request to persist changes made in the cache onto disk

cache that is used as a scratch space, and a client, in the form of a web application, that uses a REST API to send interactions to the server. Placing all the data on a hard disk allows large materials' images to be stored. However, since there may be more than one interaction required for a materials image, we employ an in-memory cache between the client and server so that interactions are only CPU-bound and are not slowed by I/O operations. We provide an explicit "Save" operation triggered from the client that allows changes introduced in the cache to be made permanent on disk. In addition, our architecture can use a name-spaced cache to safely allow for a multiple-client environment, limited only by the cache size and CPU usage needed to run the energy minimization for each interaction.

8.2 Interface

As shown in Fig. 11, the client consists of a number distinct areas that facilitate interacting with a selected materials image, obfuscating any client/server communication as much as possible.

In Fig. 11a, we present the user with a menu of tools allowing for the selection of interaction type, along with sliders to control the parameters of the addition operation in Fig. 11b. We allow slice selection by showing thumbnails of each slice in Fig. 11c, which allows the user to choose the slice to operate on, which is subsequently displayed in the designated display area shown in Fig. 11d where the interactive tools may be used. We also record log output in Fig. 11e listing actions performed by the client so that all interactive sessions may be reproduced as needed. Finally, an example annotation is shown in the display area in Fig. 11f.

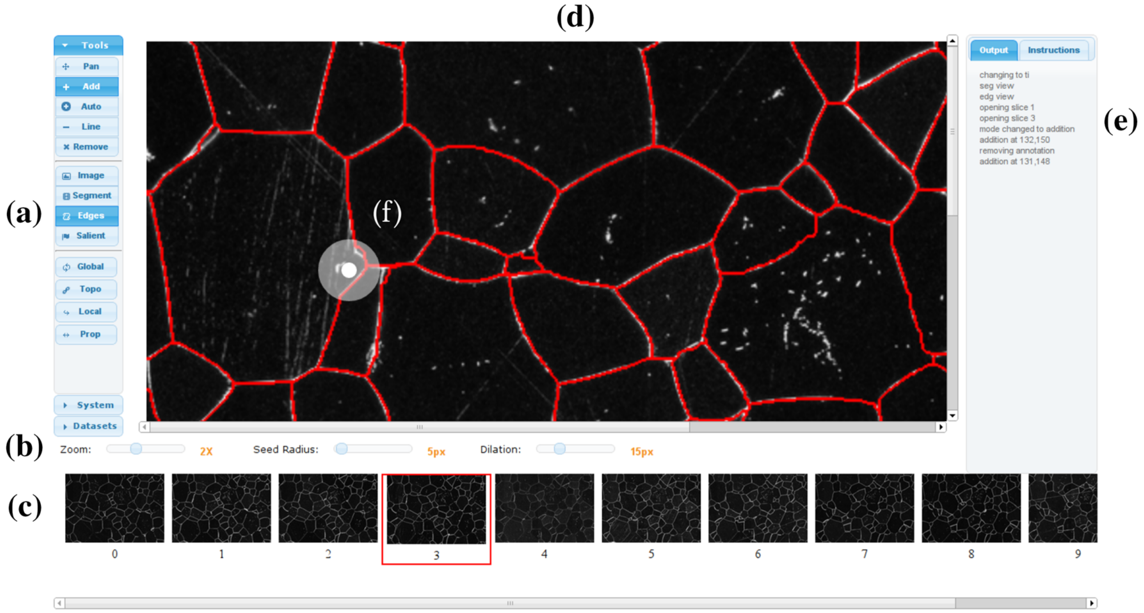
The particular tools available in the menu (Fig. 11a) are shown in Fig. 12. Specifically, we include a selection menu to choose the particular interactive tool in Fig. 12a (addition, removal, addition with parameter selection, etc.). We also include a selection menu in Fig. 12b to choose how the segmentation is displayed on the raw image, and whether the salient region identification is shown overlayed on top of the segmentation. Commands to send the annotations to the server are shown in Fig. 12c, where different commands perform different types/sequences of energy minimization (only local, global to the entire image, local with annotation repropagation, etc.). Finally, we include a menu of system tools in Fig. 12d that allow the user to erase existing annotations (reset), revert the work from the cache to the version on disk (reload), and save work from the cache to be permanent on disk (save).

An annotator's workflow consists of: (1) loading a chosen slice using Fig. 11c, (2) examining this slice using the display area in Fig. 11d, (3) choosing an appropriate annotation tool from the menu in Fig. 12a, (4) making the requisite annotation in the display area as shown by the example in Fig. 11f, (5) sending annotations to the server with the commands in Fig. 12c, and (6) saving the work, when satisfactory, with the menu item in Fig. 12c. Though we include keyboard shortcuts to accelerate this workflow, for our experiments in the next section, we explicitly follow this workflow using a mouse, and record all mouse clicks made to do the evaluation.

9 Experiments

9.1 β-Ti grains in Ti-21S

To evaluate the proposed interactive segmentation method, we use it to segment a sequence of 11 (indexed from 0



Zoom: 2X Seed Radius: 5px Dilation: 15px

```
changing to ti
seg view
edg view
opening slice 1
opening slice 3
mode changed to addition
addition at 132,150
removing annotation
addition at 131,148
```

Fig. 11 Client interface presented to the user for interaction. Consists of **a** nested menus that present interactive tools and controls, **b** sliders to control parameters, **c** slice selector **d** display area for interaction, **e** log output and instruction area, and **f** addition annotation shown displayed on the materials image

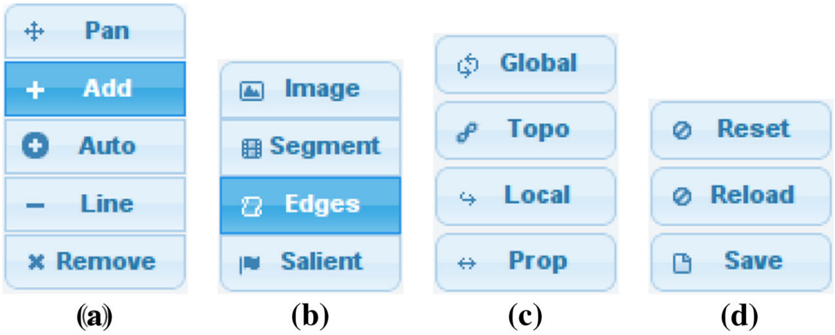
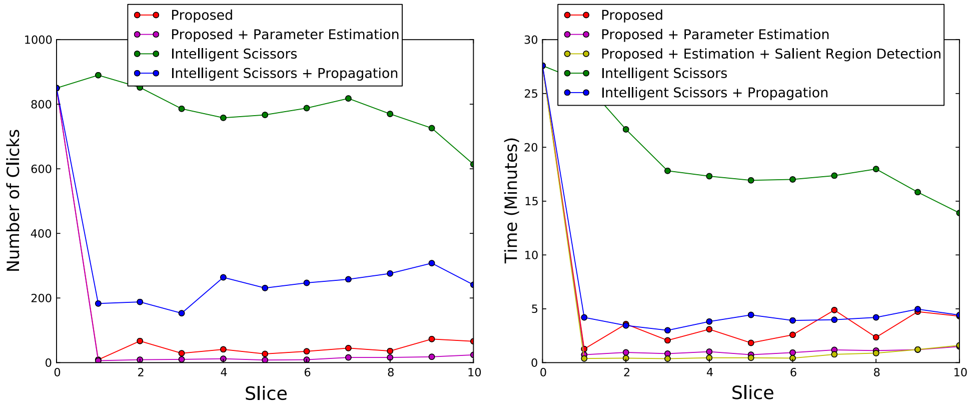


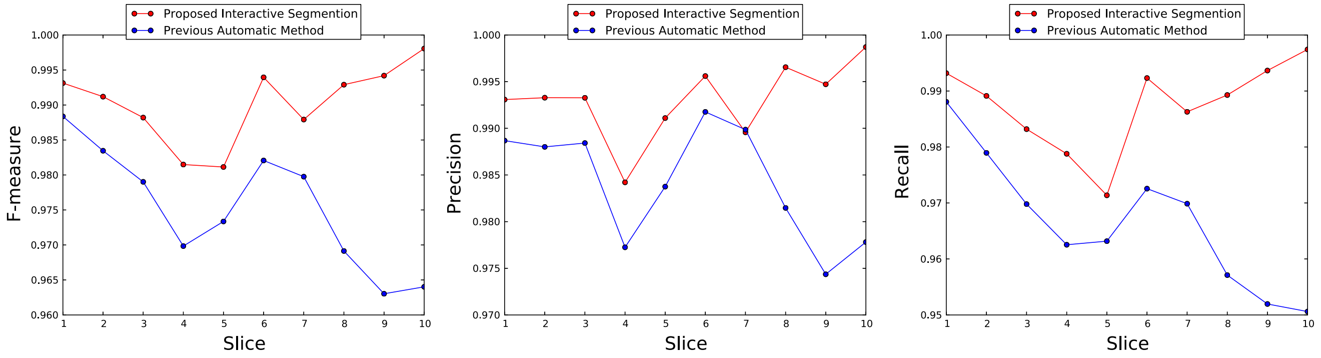
Fig. 12 Available menu groups, including **a** tool selection, **b** image display modes, **c** commands to send annotations to server, and **d** system tools to save or reload slices from the server



(a) Effort **(b)** Time

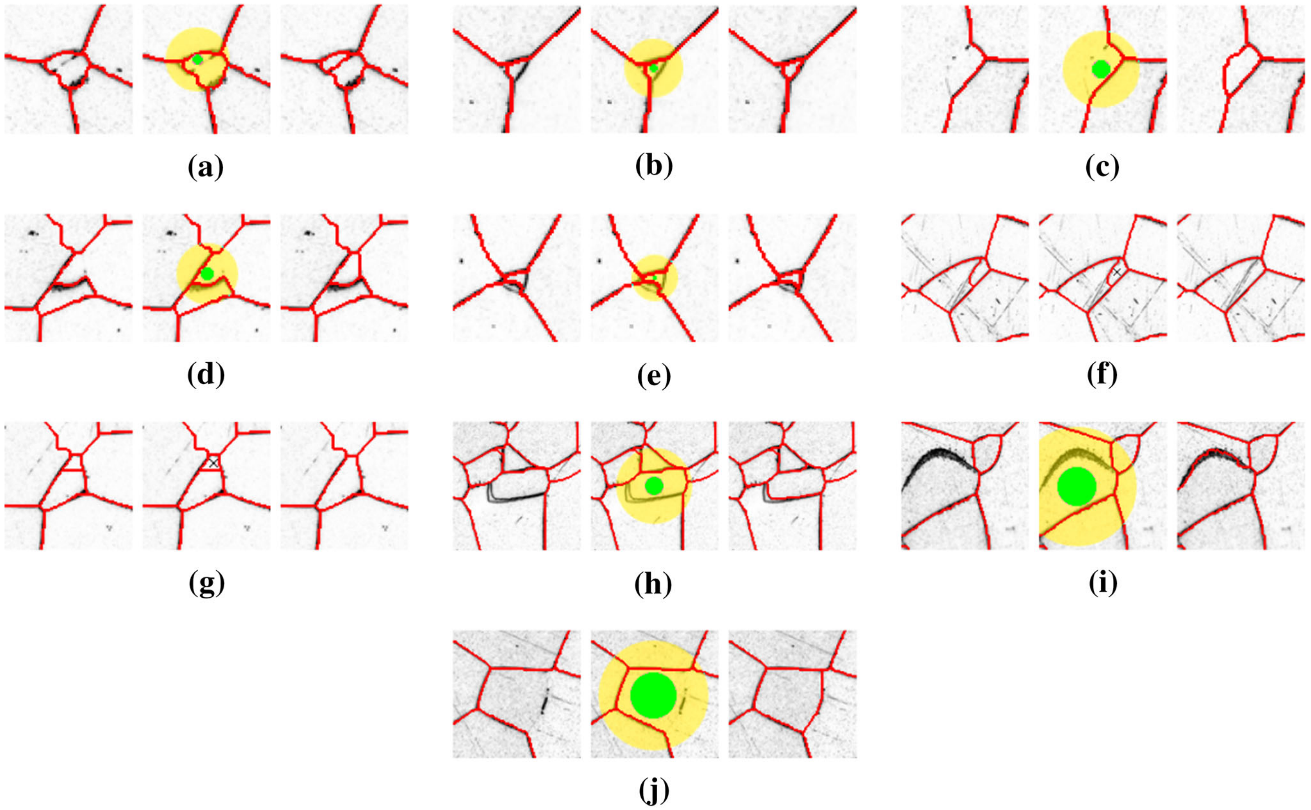
Fig. 13 Evaluation of **a** the amount of effort (number of clicks) and **b** time taken for a user to interactively segment the 11 sample slices. Smaller values are better, for both figures

to 10) microscopic titanium images [40] provided by Dave Rowenhorst, NRL. We measure the effort (i.e., number of clicks) used to segment each slice in the dataset, as well as the overall time expended by the user to segment a slice. The previous segmentation propagation approach in Sect. 2 requires a complete segmentation on one slice as an initializa-



(a) F-measure **(b)** Precision **(c)** Recall

Fig. 14 Performance of the proposed interactive segmentation method compared with our previous automated method [58] on the 11 slices, measured by the boundary coincidence with the ground truth segmentation **a** F-measure, **b** precision, **c** recall



(a) **(b)** **(c)** **(d)** **(e)** **(f)** **(g)** **(h)** **(i)** **(j)**

Fig. 15 Qualitative results where each subfigure shows the initial automatic segmentation S^V (*left*); the human annotation (*middle*) with the seed pixels in *green* and dilation pixels in *yellow*, and "X"s indicating spurious segments to be removed; and the updated segmentation $\tilde{S}^V$ (*right*). Note that **f** and **g** illustrate removal annotation and the remaining illustrate addition annotation. Image intensity inverted for clarity

tion. We count the manual segmentation on this initial slice into the effort and time required. We present the proposed method both with and without using the automatic parameter estimation discussed in Sect. 6.

For comparison, we use the readily available "intelligent scissors" interactive segmentation method [32]. Using the intelligent scissors tool, we independently segment all 11 slices from the same dataset, evaluating both effort (number of clicks) and time. In addition, we produce a hybrid of our previous automatic method [58] discussed in Sect. 2 and the intelligent scissors method, which we call "intelligent scissors + propagation" in Fig. 13. This approach uses the method from Sect. 2 to propagate a segmentation from an initial slice to the remaining slices, but it uses the intelligent scissors tool [32] to carry out the interactive component instead of the interaction proposed in this paper.

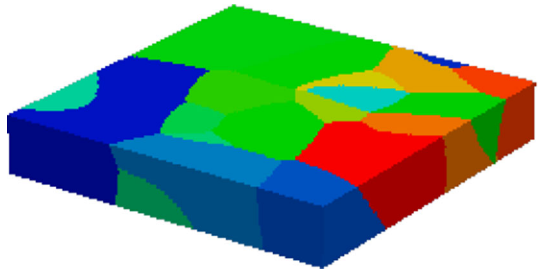


Fig. 16 Synthetic volume generated by DREAM3D [23] to evaluate the proposed interactive segmentation

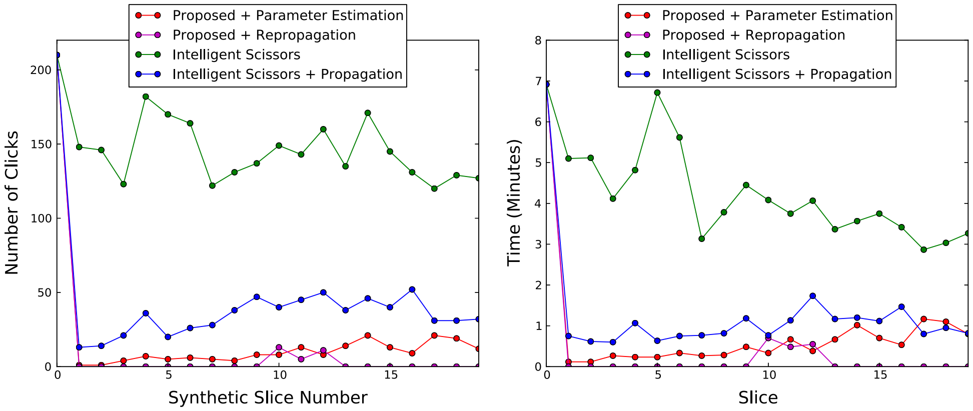
The results of this comparative experiment are shown in Fig. 13. Note that, in propagated methods ("Proposed," "Proposed + Parameter Estimation," and "Intelligent Scissors + Propagation"), the first slice is used as the initial slice U, so it requires significantly more effort and time to segment compared with the remaining slices. From Fig. 13, we can see that the method proposed in this paper ("Proposed") allows much more rapid segmentation time (<5 min in most cases) and with much less effort (<100 clicks in most cases) compared with the unpropagated intelligent scissors method. The intelligent scissors method ("Intelligent Scissors"), without the benefit of propagation, requires significantly more time and effort. The hybrid method ("Intelligent Scissors + Propagation") fares better than the unpropagated intelligent scissors method, but it still requires greater effort than the proposed method. Finally, the proposed parameter estimation method ("Proposed + Parameter Estimation") can further reduce both the time and effort required by the proposed method.

In Fig. 14, we show that the proposed interactive method is able to increase the segmentation accuracy of our state-of-the-art materials image segmentation method in Sect. 2. As in our previous work [58], we use the precision, recall, and F-measure, which is the harmonic mean of the precision and recall [31], to show the segment boundary coincidence with the manually constructed ground truth segmentation. For both the proposed and previous automatic methods, we propagate from an initial slice 0 to the remaining 10 slices, and the proposed interaction-enhanced method increases performance for all slices. Finally, qualitative segmentation results using the proposed interactive method are shown in Fig. 15 where we show the automatic segmentation results with spurious or missing segments, the human annotation, and the updated segmentation.

9.2 DREAM3D synthetic volume

We further evaluate the proposed interactive segmentation approach by generating a synthetic grain volume using DREAM3D [23]. As shown in Fig. 16, we generate a 12 µm × 12 µm × 2 µm volume where each slice is 0.06 µm × 0.06 µm × 0.1 µm in size. This volume is sampled as a 200 × 200 × 20 image, sliced along the z-axis into 20 images of size 200 × 200 each. We use these 20 synthetic slices, along with a noise model extracted from the above Ti-21S material, as additional data to evaluate the proposed interactive segmentation. As before, we evaluate interactive methods that start from an initial segmentation, and one method that does not. For methods starting from an initial segmentation, we use the results of propagating the ground truth for the first slice to all the remaining slices using the approach discussed in Sect. 2.

Quantitative results of this evaluation are shown in Fig. 17. Included in this evaluation is the proposed method with the parameter estimation discussed in Sect. 6 ("Proposed + Parameter Estimation"), the proposed method using the repropagation method discussed in Sect. 5 ("Proposed + Repropagation"), along with the intelligent scissors method used to completely segment every slice ("Intelligent Scissors") and used only to correct the propagated results ("Intelligent Scissors + Propagation"). As shown in Fig. 17a, the proposed method with parameter estimation requires less effort compared with both intelligent scissors methods. With the



(a) Effort **(b)** Time

Fig. 17 Evaluation on the DREAM3D synthetic dataset showing **a** the amount of effort (number of clicks) and **b** time taken for a user to interactively segment the 20 sample slices. Smaller values are better for both figures

addition of the repropagation method, interactions are only needed on three slices, since the corrections are then repropagated to the rest of the volume, resulting in significantly less effort than all other methods. Results for the time taken by the evaluated methods are shown in Fig. 17b, where the intelligent scissors methods all require more time, whereas the proposed method with repropagation performs better than the other evaluated methods.

10 Conclusion and future work

We have presented an interactive segmentation method extended from our automatic segmentation propagation approach. By allowing the user to interactively handle spurious and missing segments when propagating from one slice to another, we show that the time required to segment a materials image volume, as well as the overall effort (number of clicks) needed for interaction, is much less than the comparison "intelligent scissors" method used in popular image processing tools. By updating the segmentation within a local region around the interactive annotation, we are able to obtain a fast, yet robust means to handle segmentation errors when a new structure appears or an existing structure disappears from the 2D cross-section of a particular slice in the volume. We also introduce three extensions to these interactive tools: an annotation repropagation method that allows interactions to be propagated to multiple slices, a parameter estimation technique to determine the seed radius when adding a missing segment, and a salient region detection method that uses an online learning approach to guide an annotator using the interactive tools. We presented the client/server-based web application that implements these tools and show that the proposed approach, extensions, and implementation all lead to improved performance.

While this paper is focused on materials-science image segmentation for underlying grain structures, the proposed interactive method may be extended to the applications in other fields. For example, many 3D medical images [4,49,50], such as MRI and CT images, are also taken in the form of a sequence of serial-sectioned 2D slices with good structural continuity between adjacent slices. Furthermore, for some medical-imaging applications [16], the structures to be segmented are of large number with specific neighboring relations and they share high similarity to the grain structures studied in this paper. In the future, we will investigate the extension of the proposed method to improve the accuracy of medical image segmentation.

Acknowledgments This work was supported in part by AFOSR FA9550-11-1-0327 and NSF-1017199. A preliminary version of this work has been published in a conference proceedings [59].

References

1. Batra, D., Kowdle, A., Parikh, D., Luo, J., Chen, T.: iCoseg: Interactive co-segmentation with intelligent scribble guidance. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3169–3176 (2010)
2. Birkbeck, N., Cobzas, D., Jagersand, M., Murtha, A., Kesztyues, T.: An interactive graph cut method for brain tumor segmentation. In: Workshop on Applications of Computer Vision (WACV), pp. 1–7 (2009)
3. Boykov, Y., Jolly, M.: Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In: IEEE International Conference on Computer Vision, vol. 1, pp. 105–112. IEEE Press, New York (2001)
4. Boykov, Y., Jolly, M.P.: Interactive organ segmentation using graph cuts. In: Delp, S., DiGoia, A., Jaramaz, B. (eds.) Medical Image Computing and Computer-Assisted Intervention MICCAI 2000. Lecture Notes in Computer Science, vol. 1935, pp. 147–175. Springer, Berlin (2000)
5. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Trans. Pattern Anal. Mach. Intell. **26**(9), 1124–1137 (2004)
6. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. Pattern Anal. Mach. Intell. **23**(11), 1222–1239 (2001)
7. Bradski, G.: The OpenCV Library. Dr. Dobb's Journal of Software Tools (2000)
8. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. **8**(6), 679–698 (1986)
9. Cates, J.E., Lefohn, A., Whitaker, R.T.: Gist: An interactive, gpu-based level-set segmentation. Med. Image Anal. **8**(3), 217–231 (2004). http://graphics.cs.ucdavis.edu/lefohn/work/rls/tumorSeg/
10. Chuang, H., Huffman, L., Comer, M., Simmons, J., Pollak, I.: An automated segmentation for nickel-based superalloy. In: IEEE International Conference on Image Processing, pp. 2280–2283 (2008)
11. Comer, M., Bouman, C., De Graef, M., Simmons, J.: Bayesian methods for image segmentation. J. Miner. Metals Mater. Soc. **63**, 55–57 (2011)
12. Comer, M., Delp, E.: Parameter estimation and segmentation of noisy or textured images using the EM algorithm and MPM estimation. In: IEEE International Conference on Image Processing, vol. 2, pp. 650–654. IEEE Press, New York (1994)
13. Comer, M., Delp, E.: The EM/MPM algorithm for segmentation of textured images: analysis and further experimental results. IEEE Trans. Image Process. **9**(10), 1731–1744 (2000)
14. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)
15. Django Software Foundation: Django (version 1.5). http://djangoproject.com
16. Fragkiadaki, K., Zhang, W., Shi, J., Bernardis, E.: Structural-flow trajectories for unravelling tubular structure bundles. In: Medical Image Computing and Computer-Assisted Intervention, vol. 3, pp. 631–638 (2012)
17. Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 3rd edn. Prentice Hall, New York (2008)
18. Heckel, F., Konrad, O., Hahn, H.K., Peitgen, H.O.: Interactive 3D medical image segmentation with energy-minimizing implicit functions. Comput. Graph. **35**(2), 275–287 (2011)
19. Huffman, L., Simmons, J., Pollak, I.: Segmentation of digital microscopy data for the analysis of defect structures in materials using nonlinear diffusion. In: C. Bouman, E. Miller, I. Pollak (eds.) Computational Imaging VI, Proceedings of SPIE (2008)
20. Huffman, L.M., Simmons, J.P., De Graef, M., Pollak, I.: Shape priors for map segmentation of alloy micrographs using graph cuts.

In: IEEE Workshop on Statistical, Signal Processing, pp. 28–30 (2011)

21. Ibrahim, I.A., Mohamed, F.A., Lavernia, E.J.: Particulate reinforced metal matrix composites: a review. J. Mater. Sci. **26**, 1137–1156 (1991)
22. Jacinto, H., Kchichan, R., Desvignes, M., Prost, R., Valette, S.: A web interface for 3D visualization and interactive segmentation of medical images. In: 17th International Conference on 3D Web Technology (Web 3D 2012), pp. 51–58 (2012)
23. Jackson, M., Groeber, M.: DREAM3D (2012). http://dream3d.bluequartz.net
24. Jones, E., Oliphant, T., Peterson, P., et al.: SciPy: Open source scientific tools for Python (2001). http://www.scipy.org/
25. Kang, Y., Engelke, K., Kalender, W.A.: Interactive 3D editing tools for image segmentation. Med. Image Anal. **8**(1), 35–46 (2004)
26. Kiefer, W.: Intelligent scissoring for interactive segmentation of 3D meshes. Master's thesis, Princeton University (2004)
27. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? IEEE Trans. Pattern Anal. Mach. Intell. **26**(2), 147–159 (2004)
28. Kuang, Z., Schnieders, D., Zhou, H., Wong, K.Y., Yu, Y., Peng, B.: Learning image-specific parameters for interactive segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 590–597 (2012)
29. Li, Q., Ni, X., Liu, G.: Ceramic image processing using the second curvelet transform and watershed algorithm. In: IEEE International Conference on Robotics and Biomimetics, pp. 2037–2042 (2007)
30. Marroquin, J., Mitter, S., Poggio, T.: Probabilistic solution of ill-posed problems in computational vision. J. Am. Stat. Assoc. 76–89 (1987)
31. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: IEEE International Conference on Computer Vision, vol. 2, pp. 416–423 (2001)
32. Mortensen, E.N., Barrett, W.A.: Intelligent scissors for image composition. In: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '95, pp. 191–198. ACM, New York (1995)
33. Moschidis, E., Graham, J.: Propagating interactive segmentation of a single 3D example to similar images: an evaluation study using MR images of the prostate. In: IEEE International Symposium on Biomedical Imaging: From Nano to Macro, pp. 1472–1475 (2011)
34. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: machine learning in python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
35. Pfister, S.S., Betizeau, M., Dehay, C., Douglas, R.J.: INTERSEG: Interactive 3D segmentation (2012). http://n.ethz.ch/student/sabpfist/interseg.htm
36. Python Software Foundation: Python language reference. http://www.python.org
37. Reed, R.: The Superalloys: Fundamentals and Applications. Cambridge University Press, Cambridge (2006)
38. Rollett, A., Gottstein, G., Shvindlerman, L., Molodov, D.: Grain boundary mobility: a brief review. Z. Metallkunde **95**, 226–229 (2004)
39. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: Interactive foreground extraction using iterated graph cuts. ACM Transactions on Graphics (Proceedings of SIGGRAPH) **23**(3), 309–314 (2004)
40. Rowenhorst, D., Lewis, A., Spanos, G.: Three-dimensional analysis of grain topology and interface curvature in a β-titanium alloy. Acta. Mater. **58**, 5511–5519 (2010)
41. Santner, J., Pock, T., Bischof, H.: Interactive multi-label segmentation. In: Asian Conference on Computer Vision, pp. 397–410 (2011)
42. Schneider, C., Rasband, W., Eliceiri, K.: NIH image to ImageJ: 25 years of image analysis. Nat. Methods **9**, 671–675 (2012)
43. Settles, B.: Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin-Madison (2009)
44. Shalev-Shwartz, S.: Online Learning: Theory, Algorithms, and Applications. Ph.D. thesis, The Hebrew University of Jerusalem (2007)
45. Shapiro, L.G., Stockman, G.C.: Computer Vision. Prentice Hall, Upper Saddle River (2001)
46. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **22**(8), 888–905 (2000)
47. Simmons, J., Bartha, B., De Graef, M., Comer, M.: Development of bayesian segmentation techniques for automated segmentation of titanium alloy images. Microsc. Microanal. **14**(S2), 602–603 (2008)
48. Simmons, J.P., Chuang, P., Comer, M., Spowart, J.E., Uchic, M.D., De Graef, M.: Application and further development of advanced image processing algorithms for automated analysis of serial section image data. Model. Simul. Mater. Sci. Eng. **17**(2), 025,002 (2009)
49. Straehle, C., Koethe, U., Knott, G., Briggman, K., Denk, W., Hamprecht, F.: Seeded watershed cut uncertainty estimators for guided interactive segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 765–772 (2012)
50. Straehle, C.N., Köthe, U., Knott, G., Hamprecht, F.A.: Carving: scalable interactive segmentation of neural volume electron microscopy images. In: Medical Image Computing and Computer-Assisted Intervention, pp. 653–660 (2011)
51. Swiler, T.P., Holm, E.A.: Diffusion in polycrystalline microstructures. In: Annual Meeting of the American Ceramic Society (1995)
52. Tan, J., Saltzman, W.: Biomaterials with hierarchically defined micro and nanoscale structure. Biomaterials **25**(17), 3593–3601 (2004)
53. Top, A., Hamarneh, G., Abugharbieh, R.: Active learning for interactive 3D image segmentation. In: Medical Image Computing and Computer-Assisted Intervention vol. 6893, pp. 603–610 (2011)
54. Unger, M., Pock, T., Trobin, W., Cremers, D., Bischof, H.: TVSeg–interactive total variation based image segmentation. In: British Machine Vision Conference 2008, pp. 40.1–40.10 (2008)
55. Veksler, O.: Efficient graph-based energy minimization methods in computer vision. Ph.D. thesis, Cornell University, Ithaca (1999)
56. Veksler, O., Delong, A.: GCO (2011). http://vision.csd.uwo.ca/code/
57. Vezhnevets, V., Konouchine, V.: Grow-Cut–interactive multi-label N-D image segmentation. In: Graphicon, pp. 150–156 (2005)
58. Waggoner, J., Zhou, Y., Simmons, J., De Graef, M., Wang, S.: 3D materials image segmentation by 2D propagation: a graph-cut approach considering homomorphism. IEEE Trans. Image Process. **22**, 5282–5293 (2013)
59. Waggoner, J., Zhou, Y., Simmons, J., Salem, A., De Graef, M., Wang, S.: Interactive grain image segmentation using graph cut algorithms. In: Proceedings of SPIE (Computational Imaging XI), vol. 8657. Burlingame (2013)

Jarrell Waggoner is a software engineer at Groupon working in automation and demand forecasting. He received his Ph.D. from the University of South Carolina in Computer Science and Engineering. His expertise lies in computer vision and image processing applications, with work in interdisciplinary AFOSR-funded materials science image processing.

Youjie Zhou is a Ph.D. candidate in Computer Science and Engineering at the University of South Carolina (USC) working as a research assistant in the Computer Vision Lab. His main research interests include computer vision, machine learning, and large-scale multimedia analysis. He received the B.S. degree in Software Engineering from the East China Normal University (ECNU), Shanghai, China, in 2010. From 2007 to 2010, he also worked as a research assistant in the Institute of Massive Computing at ECNU, where he worked on multimedia news exploration and retrieval.

Jeff Simmons is a scientist at the Materials and Manufacturing Directorate of the Air Force Research Laboratory. He has a B.S. degree in Metallurgical Engineering from the New Mexico Institute of Mining and Technology and M.E. and Ph.D. degrees in Metallurgical Engineering and Materials Science and Materials Science and Engineering, respectively, from Carnegie Mellon University. After receiving his Ph.D. he joined the Materials and Manufacturing Directorate at Wright Patterson SFB as a post doctoral researcher and as a research contractor. In 1998, he joined the Air Force Research Laboratory as a research scientist. During his tenure, he has worked on both basic research as well as engineering applications in the aerospace industry. His research interests are in 3-D Materials Science, particularly in the development of advanced algorithms for production and analysis of large image datasets. He has published and developed programming in both the Materials Science and Signal Processing fields on application of electronic imaging algorithms towards the analysis of digital microscopy. Other research interests have included mathematical representations of microstructure and texture, physics-based modeling of microstructure development, atomistic modeling of defect properties, and computational thermodynamics. Non-research experience has included leading teams to develop analytical tools for digital data, integration of computer resources for materials science simulations, computer security, and network design as well as manufacturing particularly in machining applications. He has overseen execution of Air Force contracts on Integrated Computational Materials Science and Engineering. Dr. Simmons is a member of the TNS committees of Advanced Characterization, Testing, and Simulations, Phase Transformations, and Integrated Computational Materials Science. He is a member of TS, ACM, and the IEEE Computer and Signal Processing Societies.

Marc De Graef received a Ph.D. in Physics from the Catholic University of Leuven, Belgium, in 1989. From 1989 until 1993 he was a post-doctoral researcher in the Materials Department at the University of California at Santa Barbara. He joined the faculty of the Department of Materials Science and Engineering at Carnegie Mellon University in 1993 and is currently Full Professor and director of the J. Earle and Mary Roberts Materials Characterization Laboratory. His main research interests are in the characterization by means of X-rays and electron beams of advanced material microstructures, both in 2-D and 3-D, as well as the study of magnetic domain walls and their interactions with lattice defects by means of Lorentz transmission electron microscopy. Prof. De Graef has written more than 170 journal papers as well as two text books: *Structure of Materials* (with M.E. McHenry) and *Introduction to Conventional Transmission Electron Microscopy*, both published by Cambridge University Press. He is a Fellow of the Microscopy Society of America, and won the 2012 Educator Award from the Minerals, Metals, and Materials Society (TMS).

Song Wang received the Ph.D. degree in Electrical and Computer Engineering from the University of Illinois at Urbana-Champaign (UIUC) in 2002. From 1998 to 2002, he also worked as a research assistant in the Image Formation and Processing Group at the Beckman Institute of UIUC. In 2002, he joined the Department of Computer Science and Engineering at the University of South Carolina, where he is currently a professor. His research interests include computer vision, medical image processing, and machine learning. He is currently serving as the Publicity/Web portal Chair of the Technical Committee of Pattern Analysis and Machine Intelligence, IEEE Computer Society, and an Associate Editor of Pattern Recognition Letters. He is a senior member of IEEE and member of the IEEE Computer Society.